REMARKS

Claims 1, 3-7, 9-13, 15, 18, 20-23, and 25-27 are pending. Claims 1, 7, 15, and 18 are in independent form.

CLAIM 1

As a threshold matter, Applicant would like to address the Examiner's contention in paragraphs 4 and 4.1 of the Office action mailed April 14, 2006. In these paragraphs, the Examiner contended that the use of a text editor to change Verilog code on a server, followed by compilation of the changed code, "would appear to meet the limitations of claim 1."

Applicant respectfully disagrees. Manual editing and recompilation of Verilog or other code neither describes nor suggests elements and/or limitations in the system of claim 1. For example, to the best of applicant's knowledge, Verilog or other code does not include labels that are associated with values of signal parameters in a central database, as recited in claim 1.

This is perhaps not surprising given the nature of manual editing. In particular, with manual editing and recompilation, there is no need for a relationship between the computer code of a logic design and a central database. Rather, a programmer reads and manually updates the computer code of a logic design. No need for labels, or an association between labels and values of signal parameters, is needed. *See, e.g., Specification*, page 7, line 3-11.

Accordingly, manual editing and recompilation does not meet the limitations of claim 1.

In the action mailed April 14, 2006, claim 1 was rejected under 35 U.S.C. § 103(a) as obvious over the publication entitled "Expressive Systems," which was allegedly available at the Internet Archive Wayback Machine as of February 8, 2001 (hereinafter "Expressive") and the publication entitled "A multi-representational design data capture system," by K. Yamagishi (hereinafter "Yamagishi").

As amended, claim 1 relates to a system that includes a logic design module operable to be used by one or more users to generate a logic design as part of an electrical circuit, and a central database integrated with the logic design module and including a collection of modifiable values of signal parameters that are accessible by the logic design module. The logic design includes labels. The values of signal parameters are associated with the labels in the logic design. The logic design module is operable to update the logic design to reflect modification of the values of the signal parameters in the central database by modifying the logic design to be compatible with the modified values of the signal parameters

Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 1. For example, Expressive and Yamagishi neither describe nor suggest a logic design that includes labels, that values of signal parameters are associated with such labels in a logic design, and a logic design module that is operable to update the logic design to reflect modification of the values of the signal parameters in the central database by modifying the logic design to be compatible with the modified values of the signal parameters, as recited in claim 1.

In this regard, Expressive describes the "Expressive III," which is a hierarchy editor. *See Expressive*, page 2 under the heading "Welcome to Expressive Systems." As best understood by applicant, the hierarchy edited by Expressive III is distinct from the code of a logic design. For example, it is understood that Expressive III uses a design hierarchy to generate code. *See Expressive*, page 10 under the heading "Code Generation" (describing that Expressive is "focused on the design hierarchy" but nevertheless can be used to "automatically build a file which contains a valid compile order for design units"). *See also Expressive*, page 4, figure under the heading "Product Overview." (showing a "hierarchy partition" between Expressive III and an RTL source).

The Expressive III hierarchy editor allegedly provides various functionality for editing a hierarchy to a user, including "adding, refining signals etc." *See Expressive*, page 7 under the heading "Main User Interface." From Expressive's illustration of such editing functionality, it appears that editing is to be manual. *See Expressive*, page 8-9 under the heading "Data Dictionary." When a signal is edited, "detailed changes" can be "made and applied to the local level or the entire design." *Id.*

One specific example of manual editing described by Expressive is user-definition of signal types. *See Expressive*, page 15-16 under the heading "Types Designer." The figure on page 16 under the heading "Types Designer" is believed to illustrate the signal definition provided by Expressive. In particular, the figure on page 16 is understood to show a user editing the logical values defined for IEEE Std 1164 std_logic signals. Submitted herewith as Appendix A is a description of the IEEE Std 1164 std_logic signals. *Available at* http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/00-intro/03-stdlogic/chapter.html.

As described in Appendix A, the IEEE Std 1164 std_logic signal definitions are abstractions of the real-world states of signals that are used to simulate the behavior of a circuit. In particular, the IEEE Std 1164 std_logic characterizes individual bits in real-world signals as having nine different logical values (i.e., U, X, 0, 1, Z, W, L, H, and D) rather that the idealized two (high/low or 1/0) for bits in idealized devices. Thus, user modification of IEEE Std 1164 std_logic signal types in Expressive does not appear to change a real-world logic design. Rather, user modification of signal types in Expressive only changes the abstractions of the signals in a real-world design so that the real-world design can be modeled. Moreover, user modification of IEEE Std 1164 std_logic signal types is understood to only impact the abstraction of individual bits, rather than characteristics of multiple bit signals in a design.

When manual user input is used to "refine" a signal type, the user can select the desired user-defined (or industry standard) type for that signal. The desired user type can be "applied to a signals entire hierarchical connection automatically." *See Expressive*, page 8-9 under the heading "Data Dictionary."

Since Expressive describes that changes to hierarchy that is used to generate a model are to be made manually, there is no need for a logic design that includes labels that are associated with values of signal parameters in a central database, as recited in claim 1. Without such labels, it is self-evident that Expressive neither describes or suggests a logic design module that is operable to update the logic design to reflect modification of the values of the signal parameters in the central database by modifying the labels in the logic design to be compatible with the modified signal parameter, as recited in claim 1.

Yamagishi does not remedy these deficiencies in Expressive. Yamagishi deals with a system for capturing design data from different design representations. *See Yamagishi*, Abstract, page 13.2.1. In particular, a single base object in a common database can be used to derive the data objects for the different representations. *Id.* According to Yamagishi, design data is to be compiled incrementally into the base object of the common database. *Id.* Yamagishi does not describe or suggest that the different representations are edited in any other way than manually.

Thus, there is no need for a Yamagishi's base object (or data objects for the different representations) to include labels, much less labels that are associated with values of signal parameters in a central database, as recited in claim 1. As such, one would not be motivated to combine these references in this way.

Since neither Expressive and Yamagishi describe nor suggest elements and/or limitations recited in claim 1, claim 1 is not obvious over the cited references. Accordingly, Applicant requests that the rejections of claim 1 and the claims dependent therefrom be withdrawn.

CLAIM 7

Claim 7 was rejected under 35 U.S.C. § 103(a) as obvious over the publication entitled Expressive and Yamagishi.

As amended, claim 7 relates to a method that includes receiving an assignment of a value to a signal parameter, maintaining the value of the signal parameter in a central database in association with an identifier of the signal parameter, using the identifier of the signal parameter maintained in the central database to identify a first position in computer code for a logic design forming part of an electrical circuit, modifying the computer code at the first position to reflect the value, using the identifier of the signal parameter maintained in the central database to identify a second position in the computer code for the logic design, modifying the computer code at the first position to reflect the value, receiving an updated value of the signal parameter in the central database, and updating the first position and the second position in the computer code for the logic design to reflect the updated value of the signal parameter.

Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 7. For example, Expressive and Yamagishi neither describe nor suggest maintaining a value of the signal parameter in a central database in association with an identifier of the signal parameter and using such an identifier to identify first and second positions in computer code, as recited in claim 7.

In this regard, as discussed above, Expressive describes a systems where changes are made to a hierarchy manually. There is no use of an identifier to identify first and second positions in computer code, since Expressive is understood to limit user interaction to the hierarchy.

Yamagishi does not remedy these deficiencies in Expressive. Yamagishi deals with a system for capturing design data from different design representations. There is no mention in Yamagishi that an identifier is used to identify first and second positions in computer code.

Since Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 7, claim 7 is not obvious over the cited references. Accordingly, Applicant requests that the rejections of claim 7 and the claims dependent therefrom be withdrawn.

CLAIM 15

Claim 15 was rejected under 35 U.S.C. § 103(a) as obvious over the publication entitled Expressive and Yamagishi.

As amended, claim 15 relates to an apparatus that includes a central database accessible by one or more users, the central database including a collection of identifiers of one or more bit width signal parameters and values associated with each of the identifiers of the bit width signal parameters, modification logic to allow a user to modify the values associated with the identifiers, an interface to convey the identifiers and the associated values from the central database to a logic design module that uses the identifiers to identify where a logic design is to be changed and the values to change a bit width in the logic design to form part of an electrical circuit.

Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 15. For example, Expressive and Yamagishi neither describe nor suggest an interface to convey the identifiers and the associated values from the central database to a logic design module that uses the identifiers to identify where a logic design is to be changed and the values to change a bit width in the logic design to form part of an electrical circuit, as recited in claim 15.

In this regard, as discussed above, Expressive describes a systems where changes are made to a hierarchy manually. There is no use of an identifier to identify where a logic design is to be changed, since a user enters the changes manually. Moreover, even if one were to consider the Expressive's application of user edits to the logical values defined for IEEE Std 1164 std_logic signals to be relevant to the identification of where a logic design is to be changed (a contention with which applicant disagrees), such change to the abstraction of real-world signals does not change a logic design, much less a bit width in the logic design.

Yamagishi does not remedy these deficiencies in Expressive. Yamagishi deals with a system for capturing design data from different design representations. There is no mention in Yamagishi that an interface is to convey the identifiers and the associated values from the central database to a logic design module that uses the identifiers to identify where a logic design is to be changed and the values to change a bit width in the logic design to form part of an electrical circuit.

Since neither Expressive and Yamagishi describe nor suggest elements and/or limitations recited in claim 15, claim 15 is not obvious over the cited references. Accordingly, Applicant requests that the rejections of claim 15 and the claims dependent therefrom be withdrawn.

CLAIM 18

Claim 18 was rejected under 35 U.S.C. § 103(a) as obvious over the publication entitled Expressive and Yamagishi.

As amended, claim 18 relates to a machine-accessible medium containing instructions which cause a machine to perform operations. The operations comprise receiving a value of a signal parameter that characterizes multiple bits of a multiple bit signal, maintaining the value of the signal parameter in a central database, using the value of the signal parameter that is maintained in the central database in computer code for a logic design forming part of an electrical circuit that includes the multiple bit signal, receiving an update to the value of the signal parameter in the central database, and updating the logic design with the updated value of the signal parameter by modifying the logic design to be compatible with the updated signal parameter.

Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 18. For example, Expressive and Yamagishi neither describe nor suggest operations that include updating a logic design with an updated value of a signal parameter that characterizes multiple bits of a multiple bit signal by modifying the logic design, as recited in claim 18.

In this regard, as discussed above, Expressive's application of user edits to the logical values defined for IEEE Std 1164 std_logic signals are not updates to a logic design. Rather, these user edits are understood to only change an abstraction of the design that is used for modeling and simulation. Moreover, any such changes are understood to be relevant only to the abstraction of an individual bit, and not to a signal parameter that characterizes multiple bits of a multiple bit signal, as recited in claim 18.

Yamagishi does not remedy these deficiencies in Expressive. Yamagishi deals with a system for capturing design data from different design representations. There is no mention in Yamagishi that a logic design is to be updated with an updated value of a signal parameter that characterizes multiple bits of a multiple bit signal by modifying the logic design, as recited in claim 18.

Since Expressive and Yamagishi neither describe nor suggest elements and/or limitations recited in claim 18, claim 18 is not obvious over the cited references. Accordingly, Applicant requests that the rejections of claim 18 and the claims dependent therefrom be withdrawn.

It is believed that all of the pending claims have been addressed in this paper. However, failure to address a specific rejection, issue or comment, does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above are not intended to be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Applicant asks that all claims be allowed. No fees are believed due at this time. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: July 13, 2006

Scott C. Harris
Reg. No. 32,030

BY
JOHN F. CONROY
REG. NO. 45,485

Fish & Richardson P.C.
PTO Customer No. 20985
12390 El Camino Real
San Diego, California 92130
(858) 678-5070 telephone
(858) 678-5099 facsimile

SCH/JFC/jhg
10647397.doc